ISSN 1995-0802, Lobachevskii Journal of Mathematics, 2016, Vol. 37, No. 6, pp. 768–776. © Pleiades Publishing, Ltd., 2016.

Some New Platforms for Algebraic Cryptography and One Method of Increasing the Security

A. R. Gaynullina* **and S. N. Tronin****

(Submitted by E. K. Lipachev)

N.I. Lobachevskii Institute of Mathematics and Mechanics, Kazan (Volga Region) Federal University, Kremlevskaya ul. 18, Kazan, 420008 Tatarstan, Russia

Received June 23, 2016

Abstract—In this paper we discuss the possibility of using the categorical groupoids and the commutative operads in algebraic cryptography. Also, we introduce a general method of constructing the cryptographic protocols on the algebraic platforms. Under certain reasonable assumptions, this method allows to get a new sufficiently cryptographically strong protocol using several other protocols. Moreover, each of these protocols can be vulnerable. Sufficient cryptographic security means that the protocol will be protected for some preassigned finite time.

DOI: 10.1134/S1995080216060123

Keywords and phrases: *Algebraic cryptography, platform, cryptographic protocol, secret key, key exchange, security, computational complexity, masked group, commutative operad, masked operad, algebra over an operad, tropical semiring.*

1. INTRODUCTION

In our paper we have two main aims. The first aim is the discussion of two new platforms for algebraic cryptography. Recall that an algebraic system (a group, a ring, a field, etc.), the elements of which are used in the cryptographic protocols for the submission of some information and processing this information, is called *the platform* in algebraic cryptography (see [1, 2]). A cryptographic protocol is based on the computationally difficult problem. And often, this problem is the problem for an algebraic structure. The first new platform, which is offered in our paper, is the categorical groupoids (see [3, 4, 5]). Apparently, these objects have not been used in cryptography earlier. We discuss the categorical groupoids in Section 3. The second platform, which is discussed in our paper, is the commutative operads. The commutative operad's platforms were introduced and investigated in our previous paper [6]. Section 4 of this paper can be considered as a continuation of [6].

The second aim of our paper is the description of a new method, improving the cryptographic security. We call this method *a masking* of the algebraic system (for instance, a group, a ring, a field, etc.), that is used as a platform. This method is described in Section 2, and its specific implementations are described in Section 3 and 4. The method of masking the algebraic system allows to realize the following. Let us assume that we have some specific protocol (the encryption, or the digital signature protocol, or key exchange). And also assume that we want to modify this protocol so that it is unavailable for all possible attacks for some finite time necessary for us. We will call it, for definiteness, *the final protocol*. In fact, it is the final stage of the modified protocol. And in the first stages, there is the modification of used algebraic platform performed. As a result, the original public platform becomes a certain time public only to the protocol's participants. Thus, it becomes the secret to outsiders. Of course, this is only possible under some assumptions, which appear to us reasonable. More detailed analysis of these assumptions will be continued in our subsequent publications.

*E-mail: GaynullinaAlina@gmail.com

**E-mail: sntrnn@gmail.com

2. MASKED GROUPS

At present, algebraic cryptography develops, mostly, on the basis of the group theory (see, for example, [1, 2, 7, 8]). Thus, we begin by discussing the case of a group platform.

Let G be a group with the multiplication xy and with the fixed element $c \in G$. We define a new multiplication on the set G by $x \cdot y = xcy$. Then $(G, \cdot)$ is a group with the identity element c^{-1}. The element $c^{-1}x^{-1}c^{-1}$ is the inverse element of the element x. Denote the resulting group by G_c. We call it *a masked group with the masking element* c.

It is known that $G \cong G_c$. This isomorphism is given by the correspondence $x \mapsto c^{-1}x$.

The meaning of the transition from G to G_c for cryptography is as follows. Suppose we have some cryptosystem (the encryption, or the digital signature, or the creation of a common secret key) on the platform of the group G. Then there is a public description of the group G, suitable for the deployment of G elements in computer memory, and for the convenient processing of the data. The group's elements are usually represented by bit strings.

To be specific let us assume that the group G is defined by a finite set of generators and relations. At the same time, it is required that the elements of G should have a normal form. The normal form is the way of writing some element as a word composed of the generators and their inverses. Moreover, different elements must have different normal forms. Suppose that the fixed bit strings are compared with the generators and their inverses. Then the element in the normal form is associated with a string concatenation which corresponds to multipliers. The multiplication of the elements in the normal form involves two stages. In the first stage two words are written to each other. This action corresponds to the concatenation of strings of the multipliers. In the second stage the resulting word is transformed into a normal form. Thus, there must be an algorithm that processes an arbitrary word, or more precisely, the product of the generators and their inverses, into a normal form. This algorithm solves the word problem. Moreover, this algorithm must be an efficient algorithm. It is known that only the groups with the algorithm of this kind are suitable for cryptography.

Suppose now that the protocol's participants created a shared secret masking element, and also suppose that they implement the protocol in the masked group. It means that there are some changes of the following data: the normal form of the elements, the relations and the algorithm, converting the words to the normal form. The masking element is converted into the secret parameter. Externally the coding method of the generators by bit strings remains the same. However, the bit strings corresponding to other elements are changed. An attacker does not know the element c. Herewith the method of changing is unknown to the attacker. Note that if you don't know c, it will be impossible to carry out any action in G_c. Thus, an attack on the protocol is impossible.

Let P be some auxiliary protocol of the creation of a shared secret key. Then the protocol's participants compute the secret masking element c using the protocol P. Denote by $c = P(G)$ the result of the protocol P.

We accept the following assumption. Let t be a runtime of the protocol P (more precisely, the upper limit for all possible particular implementations). Let us assume that there is an attack to P, which allows to find c. Also, assume that the runtime of the attacking algorithm is not less than T in all possible cases. Our main assumption is the following inequality: $t < T$. Let $\Delta = T - t > 0$. Then it is possible to implement the following cyclic process:

$$
\begin{aligned}
&\textit{Step } 1. && c_1 = P(G), && G_1 = G_{c_1};\\
&\textit{Step } 2. && c_2 = P(G_1), && G_2 = (G_1)_{c_2};\\
&\quad\vdots && \quad\vdots && \quad\vdots\\
&\textit{Step } n-1. && c_{n-1} = P(G_{n-2}), && G_{n-1} = (G_{n-2})_{c_{n-1}};\\
&\textit{Step } n. && c_n = P(G_{n-1}). &&
\end{aligned}
$$

Next the protocol is implemented in the masked group $(G_{n-1})_{c_n}$.

At each step of this cyclic process, the attacker lags behind for the time Δ. Thus, the attacker lags behind for the time $n\Delta$ within n steps. Herewith the, protocol's participants may choose the number n arbitrarily large. The protocol of the creation of a shared secret key includes a random selection of

some parameters. Therefore, the attacker can not jump even through one step. To find the final masking element, he will have to spend at least time Tn. The protocol's participants will compute the masking element c_n at most for the time tn.

As a result, it may take time, at least Δn, until the attacker can attack the protocol. During this time, the protocol will be protected. Thus, there is a theoretical possibility of constructing the fairly good protocols even from the vulnerable components. In our understanding, good enough protocols are the protocols, providing the cryptographic security for some end, but sufficiently long time, and this time can be adjusted by the protocol's participants.

The conditions of performing the inequality $t < T$, will be investigated in our consequent papers. It is connected with the study of the lower limits of the complexity of some particular algorithms. It is known that many secret key exchange protocols are based on the complexity of these algorithms. Thus, it is expected that the lower bounds of the complexity is also very considerable.

As well as the masking of groups, we can mask any associative ring R with an identity. Select the masking element c in the group $U(R)$ of invertible elements of the ring. Next, construct new masked ring R_c, which coincides with R as an Abelian group. But on R_c we define new multiplication as $x \cdot y = xcy$.

As a result, we obtain an associative ring with the identity c^{-1}, and $U(R_c) = U(R)$ again. Clearly, the rings are isomorphic: $R \cong R_c, x \leftrightarrow c^{-1}x$. Previous discussions for the groups can be repeated to the rings almost word for word. More precisely, we can provide sufficient cryptographic security of the protocols on a platform of some "good" rings. Possible requirements for rings are as follows. Firstly, the group $U(R)$ should be large enough, and secondly, there must be a good algorithm for selecting a random invertible element of a ring.

To increase the cryptographic security of group-based cryptosystems, we can use more than one group, namely, the large parameterized family of groups. Let I be some large family of parameters, and for any $i \in I$ there is given the group $G(i)$. We can assume that these groups are generally known (not secret). Suppose now that the protocol's participants choose the shared secret parameter $j \in I$ in some way. Then they will be able to perform the protocol in the group $G(j)$. Herewith $G(j)$ will be an unknown to the attacker. Of course, this will require some additional conditions. One of these conditions may be as follows. Assume that there is the group G given and for any $i \in I$, the element $c_i \in G$ is given too. Then $G(i) = G_{c_i}$. The elements c_i can be computed by some quite easy rule. For instance, assume $I = \{1, 2, \ldots, m\}$, and also assume that there is given the element $c \in G$, where the order of c is strictly less than m. Then we can put $c_i = c^i$. So, $G(i) \cong G$ for any i, but the multiplication in $G(i)$ is arranged as $x \cdot y = xc_iy$. Suppose we need to perform some protocol on the platform G. In this case, the protocol's participants previously choose the shared secret parameter $j \in I$, and then they perform the protocol in the group $G(j) = G_{c_j}$. As noted above, an attack on the protocol will not be possible, until the attacker computes j, and finds c_j. Hence, in particular, we obtain economical use of the memory.

We can enhance the degree of protection as follows. Let us assume that originally there is given a family of the public elements c_j. Then for each of them it is possible to apply the above cyclic process. Thus, we can get the secret for the attacker the parameter c_j^*. After that, all parameterized family $G^*(j) = G_{c_j^*}$ becomes the secret to the attacker. In order to exclude brute-force search, the set I should be large. More specifically, if, for example, $c_j = c^j$, then it will be sufficient to apply the above cyclic process to the element c. Put $c_1 = c$, and further continue as above. Of course, this is not the only option to create the difficulties for the attacker. Difficulties in this case are necessary to spend a lot of time before it will be possible to attack the final protocol. Again, we assume that the attacker can attack all parts of the upgraded protocol. And we also assume that it attacks on every step of the protocol are slower than every step of the protocol are performed.

3. CATEGORICAL GROUPOIDS

In this section, we describe another new platform for algebraic cryptography, namely, categorical groupoids. Note that the categorical groupoids and the groupoids are different mathematical objects. Recall some necessary definitions (see [9]).

Definition 1. *A category K consists of the following data:*

1) *a class $Ob(K)$ of objects (K-objects);*

2) *for any two objects $i, j \in Ob(K)$, a set $K(i, j)$ of morphisms with domain i and codomain j;*

3) *for any object $i \in Ob(K)$, an identity morphism $1_i : i \to i$;*

4) *for any objects $i, j, k \in Ob(K)$, and for any morphisms $\alpha : i \to j, \beta : j \to k$, a function called composition $K(j, k) \times K(i, j) \to K(i, k), (\beta, \alpha) \mapsto \beta\alpha$. Call the morphism $\beta\alpha$ the composition of β and α.*

The above data is required to satisfy the following associativity and unity axioms.

Associativity. Suppose $\alpha : i \to j, \beta : j \to k, l \in Ob(K)$, and $\gamma \in K(k, l)$. Then there is an equality $\alpha(\beta\gamma) = (\alpha\beta)\gamma$ in $K(i, l)$.

Unity. For any objects $i, j \in Ob(K)$, there are equalities: $\alpha 1_i = \alpha = 1_j\alpha$ in $K(i, j)$ for all $\alpha \in K(i, j)$.

We do not assume that the reader has any prior knowledge of category theory. The book [9] have all the category theory that we will need in this paper.

Note that a monoid is a category K with one object i, and with a set $K(i, i)$ of morphisms.

Definition 2. *Two objects i and j are isomorphic in the category K if there are the morphisms $\alpha : i \to j, \beta : j \to i$ such that $\beta\alpha = 1_i$, and $\alpha\beta = 1_j$. Morphisms α, β are called isomorphisms; we write $\beta = \alpha^{-1}$.*

Definition 3. *If A and B are categories, then a functor F from A to B is a function that assigns to each A-object i a B-object $F(i)$, and to each A-morphism $f : i \to j$ a B-morphism $F(f) : F(i) \to F(j)$, in such a way that*

1) *F preserves composition; i.e., $F(fg) = F(f)F(g)$ whenever fg is defined, and*

2) *F preserves identity morphisms; i.e., $F(1_i) = 1_{F(i)}$ for each A-object i.*

Definition 4. *A category whose objects form a set and in which every morphism is an isomorphism is called a categorical groupoid.*

For brevity, we use "a groupoid" instead of "a categorical groupoid". For example, a group, regarded as a category with one object, is also a groupoid. Groupoids similar to parameterized set of groups from Section 2. Indeed, if K is groupoid, then for any object i the set $K(i, i) = K(i)$ is group under composition of morphisms.

Note that if there is the morphism $\alpha : i \to j$ then we provide an isomorphism between $K(i)$ and $K(j)$ as follows: $\gamma \mapsto \alpha\gamma\alpha^{-1}$, where $\gamma : i \to i$ and $\alpha\gamma\alpha^{-1} : j \to j$. Furthermore, there is an one-to-one correspondence between $K(i, i)$ and $K(i, j) : \gamma \mapsto \alpha\gamma$, where $\gamma : i \to i$ and $\alpha\gamma : i \to j$.

The groupoid's theory can be found in [3, 4, 5]. Nowadays the groupoids are used mainly in algebraic topology.

Definition 5. *A groupoid K is called connected if $K(i, j)$ is non-empty for all objects i, j of K.*

Further we consider only connected groupoids.

Let us describe an example of a groupoid, which can be used in cryptography. Let G be a group, and let I be a set. The elements in the set I are called the objects of groupoid K. Assume $K(i, j) = G$ for any i, j. Take the element $c_i \in G$ for any $i \in I$. Then for $\alpha : i \to j, \beta : k \to i$, we can define the composition by $\alpha\beta = \alpha c_i \beta$. Note that on the right-hand side of this equality there is the product in the group G. It is easy to verify that in this case the properties of a category are performed, and $1_i = c_i^{-1}$. Clearly, $\alpha^{-1} = c_i^{-1}\alpha^{-1}c_j^{-1}$. Therefore, this category is a groupoid.

The described groupoid is suitable for use in cryptography. Let us consider the following example. Assume that we want to modify (increase the cryptographic security) some final protocol, which was originally defined in the group platform. We transfer this protocol to the groupoid's platform. Public data are the group G and the set of the elements $c_i \in G, i \in I$, where I is a large set. Hence there is the groupoid K defined, and it is public. We assume for simplicity that $c_j = c^j, I = \{1, 2, \ldots, m\}$ for some large m, and also assume that the order of $c \in G$ is less than m. Firstly, according to the cyclic process from Section 2, we obtain from $c_1 = c$ the secret element $c^* = c_n$. Then $c_j^* = (c^*)^j$, and new groupoid

K^* is constructed using these data as above. All compositions of morphisms in the groupoid K^* are known only by the protocol's participants. Hence this groupoid is not public. At the same time, the attacker spends much time to compute the element c^*.

The next step of the modified protocol is to select the public object i, the common secret object j, and the common secret morphism $\alpha : i \to j$. It can be done by using some protocols of selection the shared secret key.

Then the protocol's participants use the group $K_i^* = K^*(i,i)$, and generate the input data for the final protocol. And then they publish such of them, which are public. Note that if in the protocol there are a hash function, or some other function defined on the group G, or with values in G, then these functions are trivially redefined as follows. Group G is replaced to the group $K(i,i)$ (or $K(j,j)$). This is the same group.

Next, if g is some public (or secret) element of $K^*(i,i)$, then it is mapped to $K^*(j,j)$ by the isomorphism $g \mapsto \alpha g \alpha^{-1}$.

As α is secret, then the initial public element g is mapped to an element that is known only by the protocol's participants. As a result, all group data of the original protocol are isomorphically mapped to the secret group $K^*(j,j)$ using a secret isomorphism. Thus, data, initially public to an attacker, are minimized. Of course, we assume that each pick of a shared secret key can be attacked. But we also assume that the attack time is longer than the execution time of the protocol.

At the last step of the modified protocol, the final protocol is performed in the group $K^*(j,j)$, and most of the initial data (including the group structure) are known only by the protocol's participants.

Thus, from a few protocols, namely, one final protocol and few protocols of the creating a shared secret key, we can assemble a new protocol. This protocol solves the same problem as the final protocol, but it has some certain guarantees of the cryptographic security. These guarantees will be valid, even if all protocols, which made the assembly, are attackable to some extent.

Above we constructed a groupoid using the group G and the set of its elements $c_x, x \in X$. Denote this groupoid by $(G; X; \{c_x | x \in X, c_x \in G\})$.

Theorem 1. *Let $\mathcal{G}$ be a connected groupoid, and let $X = Ob(\mathcal{G})$ be a set. Then $\mathcal{G}$ is isomorphic to the groupoid $\mathcal{G}_0 = (G; X; \{c_x | x \in X, c_x \in G\})$.*

Proof. Let us fix $x_0 \in X$. Assume that $G = \mathcal{G}(x_0, x_0)$. For any $x \in X$ pick $b_x : x_0 \to x$, and $a_x : x \to x_0$. Let $c_x = a_x b_x$.

We construct the functor $\Phi : \mathcal{G}_0 \to \mathcal{G}$. This functor maps any object x to x, $\Phi(x) = x$.

Let $g : x \to y$ be a morphism of $\mathcal{G}_0(x,y)$. In fact, it is an element of G. The map $\mathcal{G}_0(x,y) \to \mathcal{G}(x,y), g \mapsto \Phi(g)$ is constructed as $\Phi(g) = b_y g a_x$. Let us prove that Φ is functor. Assume that $g : x \to y, h : z \to x$ are morphisms of $\mathcal{G}_0$. In fact, $g, h \in G$. Then $g \cdot h : z \to y$ is $g c_x h = g a_x b_x h$. Clearly, $\Phi(g \cdot h) = \Phi(g a_x b_x h) = b_y g a_x b_x h a_z$, $\Phi(g)\Phi(h) = (b_y g a_x)(b_x h a_z)$. Thus, $\Phi(g \cdot h) = \Phi(g)\Phi(h)$.

The identity morphism of the object x in $\mathcal{G}_0$ is $c_x^{-1} = b_x^{-1} a_x^{-1}$. Then $\Phi(b_x^{-1} a_x^{-1}) = b_x (b_x^{-1} a_x^{-1}) a_x = 1_x$ is the identity morphism of the object x in $\mathcal{G}$.

Thus, the functor Φ is constructed.

Let us consider the inverse functor $\Psi : \mathcal{G} \to \mathcal{G}_0$. The map $\mathcal{G}(x,y) \to \mathcal{G}_0(x,y), u \mapsto \Psi(u)$ is constructed as: $\Psi(u) = b_y^{-1} u a_x^{-1}$.

Clearly, this map is inverse to the constructed above map Φ. Verify that if $u : x \to y, v : z \to x$ are morphisms of $\mathcal{G}$, then $\Psi(uv) = \Psi(u) \cdot \Psi(v) = \Psi(u) c_x \Psi(v)$.

Indeed, $\Psi(uv) = b_y^{-1} uv a_z^{-1}$, $\Psi(u) c_x \Psi(v) = (b_y^{-1} u a_x^{-1}) a_x b_x (b_x^{-1} v a_z^{-1})$, $\Psi(1_x) = b_x^{-1} 1_x a_x^{-1} = c_x^{-1}$. This completes the proof of the theorem. □

Consequence 1. *Any connected groupoid is isomorphic to the groupoid of type $(G; X; c_x)$, where $c_x = 1_G$ is the identity element of the group G for any x.*

4. COMMUTATIVE OPERADS

In [6] we show the use of commutative operads in public-key cryptography. Assume that the definitions, the notations, and the terminology of [6] are known. The general theory of operads can be found, for instance, in [10–12]. Commutative operads were introduced in [13] and [14]. In [15] we studied one class of commutative operads, namely, the operads of multidimensional cubes in Euclidean spaces and their generalization.

Let R be an operad. Then $R(1)$ is a semigroup with an identity element. Take some invertible element $c \in R(1)$. Then we can construct the masked operad R_c analogously to the masked groups in Section 2. Namely, we introduce a new operad composition:

$$R_c(m) \times R_c(n_1) \times \cdots \times R_c(n_m) \to R_c(n_1 + \cdots + n_m),$$
$$(w, w_1, \ldots, w_m) \mapsto [ww_1 \ldots w_m]_c.$$

Assume

$$[ww_1 \ldots w_m]_c = w(cw_1) \ldots (cw_m) = (w\underbrace{c \ldots c}_{m})w_1 \ldots w_m. \tag{1}$$

Theorem 2. *R_c with the operation of composition* (1) *is an operad. The element $c^{-1} \in R(1)$ is the identity element of this operad. If the operad R is commutative, then the operad R_c is also commutative. The operads R* and *R_c are isomorphic. The isomorphism is given by the set of mappings $\psi_n : R(n) \to R_c(n), \psi_n(w) = cw$.*

Remark. The expression cw is the result of operadic composition $R(1) \times R(n) \to R(n)$. The expression $wc \ldots c$ is the result of operadic composition $R(m) \times R(1) \times \cdots \times R(1) \to R(m)$.

Proof. By definition. □

Let A be *an algebra* over the operad R. It means that for all $n \geq 0$ a set A is endowed with some mappings of the form:

$$R(n) \times A^n \to A, \qquad (w, a_1, \ldots, a_n) \mapsto wa_1 \ldots a_n,$$

that satisfy certain properties (see [6] for more details).

Let us construct the masked algebra A_c over the operad R_c as follows. Let $A_c = A$ as a set. We determine the operations of composition as follows:

$$R_c(n) \times A_c^n \to A_c, \quad (w, a_1, \ldots, a_n) \mapsto [wa_1 \ldots a_n]_c,$$

where

$$[wa_1 \ldots a_n]_c = w(ca_1) \ldots (ca_n) = (wc \ldots c)a_1 \ldots a_n. \tag{2}$$

Theorem 3. *The operations* (2) *determine the structure of an algebra over the operad R_c on A_c. The varieties* $\mathrm{Alg}(R)$ *and* $\mathrm{Alg}(R_c)$ *are rationally equivalent* (*see* [16, Definition 1.2.2, p. 26]). *If $F : \mathrm{Alg}(R) \to \mathrm{Alg}(R_c)$ is the functor that implements the equivalence, then F is constructed as $F(A) = A_c$.*

Proof. Direct verification. □

Now we can apply the method of Section 2 to increase the cryptographic security for all protocols from [6]. This method works, if the group of invertible elements $R(1)$ is large enough. It eliminates the possibility of iterate of the masking elements.

Assume $Z = \{Z(n) | n \geq 1\}$, where $Z(n) = K^n$. An element $Z(n)$ is a sequence (string) $\bar{x} = (x_1, \ldots, x_n)$ of elements $x_i \in K$. The action of an element $g \in K$ on the string $\bar{x}$ is defined as follows $g\bar{x} = (gx_1, \ldots, gx_n)$. The composition in this operad is defined as:

$$Z(m) \times Z(n_1) \times \cdots \times Z(n_m) \to Z(n_1 + \cdots + n_m),$$

$$(\bar{x}, \bar{y}_1, \ldots, \bar{y}_m) \mapsto \bar{x}\bar{y}_1 \ldots \bar{y}_m,$$

where $\bar{x} = (x_1, \ldots, x_m) \in R(m)$, $\bar{y}_i = (y_{i,1}, \ldots, y_{i,n_i}) \in R(n_i)$ for all $1 \leq i \leq m$, and $\bar{x}\bar{y}_1 \ldots \bar{y}_m = (x_1\bar{y}_1, \ldots, x_m\bar{y}_m)$, $x_i\bar{y}_i = (x_iy_{i,1}, \ldots, x_iy_{i,n_i})$.

The permutation group Σ_n acts on a set $Z(n)$ as follows: $(x_1,\dots,x_n)\sigma=(x_{\sigma(1)},\dots,x_{\sigma(n)})$.

Example of an R-algebra from [6] is generalized as follows. Let K be an associative commutative ring or semiring, Z be a commutative operad.

Consider an arbitrary suboperad R of the operad Z, and determine the structure of R-algebra on $A=K^m$. Let $k_1,\dots,k_m$ be fixed positive integers, $y_1,\dots,y_m\in K$. We define the mappings:

$$R(n)\times A^n\longrightarrow A,\qquad \xi a_1\dots a_n=(b_1,\dots,b_m),$$

where $\xi=(x_1,\dots,x_n)\in R(n)$, $a_i=(a_{1,i},\dots,a_{m,i})$, $1\le i\le n$, and

$$b_i=\sum_{j=1}^{n}a_{i,j}x_j^{k_i}y_i \tag{3}$$

for all i, $1\le i\le m$.

Lemma 1. *The equality* (3) *defines the structure of R-algebra on A^m.*

Proof. Straightforward check. □

Denote this algebra by $A(k_1,\dots,k_m;y_1,\dots,y_m)$, or briefly, by $A(\{k_i\};\{y_i\})$.

Next, let us recall Protocol 1 from [6]. Recall also that $\sum_{i=1}^{n}{}^{(\xi)}a_i=\xi a_1\dots a_n$.

Protocol 1. The creation of a shared secret key

Alice's secret is $\omega\in R(n)$. Bob's secret is $\lambda\in R(m)$. Public elements are $a_{i,j}\in A$, $1\le i\le n$, $1\le j\le m$.

1) Alice computes $\alpha_j=\sum_{i=1}^{n}{}^{(\omega)}a_{i,j}$, $1\le j\le m$.

2) Bob computes $\beta_i=\sum_{j=1}^{m}{}^{(\lambda)}$, $a_{i,j}$, $1\le i\le n$.

3) Alice sends the elements α_j to Bob.

4) Bob sends the elements β_i to Alice.

5) Finally, Alice computes $\sum_{i=1}^{n}{}^{(\omega)}\beta_i$, and Bob computes $\sum_{j=1}^{m}{}^{(\lambda)}\alpha_j$.

By definition of a commutative operad, $\sum_{i=1}^{n}{}^{(\omega)}\sum_{j=1}^{m}{}^{(\lambda)}a_{i,j}=\sum_{j=1}^{m}{}^{(\lambda)}\sum_{i=1}^{n}{}^{(\omega)}a_{i,j}$. Thus, Alice and Bob receive a shared secret key.

The security of Protocol 1 is based on the complexity of the task of finding $\xi\in R(k)$ using known $b_1,\dots,b_k\in A$ and $\sum_{i=1}^{k}{}^{(\xi)}b_i\in A$.

Next, we need a generalization [6, Example 7], where the operad R is based on the tropical semiring K. Recall that in this case, $K=\mathbb{Z}$ as a set, addition is defined by $a\oplus b=\min(a,b)$, and multiplication is defined by $a\odot b=a+b$. Note that 0 is the unit of this semiring. Tropical semiring was first used in cryptography in [17].

According to [6, Theorem 2], the cryptographic security of Protocol 1 for $A=A(\{k_i\};\{y_i\})$ depends on the complexity of solving a large system of equations of the type

$$\begin{cases}(a_{1,1}\odot x_1^{\odot k_1})\odot y_1 \;\oplus\; \cdots \;\oplus\; (a_{1,n}\odot x_n^{\odot k_1})\odot y_1 \;=\; b_1;\\ \cdots\quad\cdots\quad\cdots\quad\cdots\\ (a_{m,1}\odot x_1^{\odot k_m})\odot y_m \;\oplus\; \cdots \;\oplus\; (a_{m,n}\odot x_n^{\odot k_m})\odot y_m \;=\; b_m\end{cases} \tag{4}$$

over a tropical semiring K.

In the standard notation, the system (4) is as follows:

$$\begin{cases}\min\limits_{1\le i\le n}(a_{1,i}+k_1x_i+y_1)=b_1;\\ \cdots\qquad\cdots\\ \min\limits_{1\le i\le n}(a_{m,i}+k_mx_i+y_m)=b_m.\end{cases}$$

Suppose that the group of invertible elements of the semigroup $R(1)$ is large enough. Then we have a possibility to mask the operad R and the algebra $A(\{k_i\};\{y_i\})$. The way of implementation of it for the operad, is described above.

In the case of an algebra, we need to do all parameters $k_1,\dots,k_m$, $y_1,\dots,y_m$ the secret. It can be done by using some protocol of the selection of a shared secret key.

We take the masking element $c\in R(1)=\mathbb{Z}$. Then the composition in R_c takes the form:

$$R_c(m)\times R_c(n_1)\times\cdots\times R_c(n_m)\to R_c(n_1+\cdots+n_m),$$

$$(w,u_1,\dots,u_m)\mapsto [wu_1\dots u_m]_c,$$

where $w=(w_1,\dots,w_m)$, $u_i=(u_{i,1},\dots,u_{i,n_i})$.

We have

$$[wu_1\dots u_m]_c=(w_1\odot c\odot u_{1,1},\dots,w_1\odot c\odot u_{1,n_1},w_2\odot c\odot u_{2,1},\dots,w_2\odot c\odot u_{2,n_2},\dots)$$

Note that the equality $w_i\odot c\odot u_{i,j}=w_i+c+u_{i,j}$ is true in the semiring K.

The algebra A_c is given by the following map:

$$R_c(n)\times A_c^n\to A_c,\quad (w,a_1,\dots,a_n)\mapsto [wa_1\dots a_n]_c.$$

Assume that $w=(w_1,\dots,w_n)$, $a_j=(a_{1,j},\dots,a_{m,j})$, $1\le j\le n$. Then $[wa_1\dots a_n]_c=(z_1,\dots,z_m)\in A_c=K^m$, and

$$z_i=a_{i,1}\odot c^{\odot k_i}\odot w_1^{\odot k_i}\oplus\cdots\oplus a_{i,n}\odot c^{\odot k_i}\odot w_n^{\odot k_i},$$

where $1\le i\le m$, $1\le j\le n$.

In the case of $A=A(\{k_i\};\{y_i\})$, we get the following result.

Theorem 4. *The cryptographic security of Protocol* 1 *for the operad R_c and for the algebra $A(\{k_i\};\{y_i\})$ depends on the complexity of solving a system of tropical equations of the type:*

$$\bigoplus_{j=1}^{n}(a_{i,j}\odot c^{\odot k_i}\odot x_j^{\odot k_i}\odot y_i)=b_i,\quad 1\le i\le m \tag{5}$$

where $x_1,\dots,x_n$, $k_1,\dots,k_m$, $y_1,\dots,y_m$, and c are unknown.

In the standard notation, the system (5) *is as follows:*

$$\min_{1\le j\le n}(a_{i,j}+k_ic+k_ix_j+y_i)=b_i,\quad 1\le i\le m.$$

Proof. By definition and by [6, Theorem 2]. □

Note that, in the general case, the system of equations (5) is significantly nonlinear. We are not aware the methods for solving the systems of equations of this type. Thus, we expect that Protocol 1 on a tropical semiring has sufficient cryptographic security.

REFERENCES

1. A. Myasnikov, V. Shpilrain, and A. Ushakov, *Non-commutative Cryptography and Complexity of Group-theoretic Problems* (American Mathematical Society, Providence, Rhode Island, 2011).
2. V. A. Romankov, *Algebraic Cryptography* (Omsk University Publishing House, Omsk, 2013) [in Russian].
3. P. J. Higgins, *Categories and Groupoids*, Reprints in Theory and Applications of Categories, No. 7, 1–195 (1971).
4. R. Brown, *Topology and Groupoids* (Booksurge LLC, S. Carolina, 2006).
5. R. Brown, P. J. Higgins, and R. Sivera, *Nonabelian Algebraic Topology* (EMS, Switzerland, 2011).
6. A. R. Gaynullina and S. N. Tronin, "Towards an Operad-Based Cryptography: Applications of Commutative Operads," Lob. J. Math. **37** (3), 234–239 (2016).
7. G. Baumslag, B. Fine, M. Kreuzer, and G. Rosenberger, *A Course in Mathematical Cryptography* (Walter de Gruyter GmbH, Berlin/Boston, 2015).
8. M. I. G. Vasco and R. Steinwandt, *Group Theoretic Cryptography* (CRC Press, Taylor & Francis Group, Boca Raton, London–New York, 2015).
9. S. Mac Lane, *Categories for the Working Mathematician*, 2nd ed. (Springer-Verlag, New York, 1998).

10. M. Markl, S. Shnider, and J. Stasheff, *Operads in Algebra, Topology and Physics*, Math. Surveys and Monographs, Vol. 96 (AMS, USA, 2002).
11. J.-L. Loday and B. Vallette, *Algebraic Operads* (Springer-Verlag, Berlin Heidelberg, 2012).
12. M. R. Bremner and V. Dotsenko, *Algebraic Operads. An Algorithmic Companion* (CRC Press, Taylor & Francis Group, Boca Raton, FL, 2016).
13. S. N. Tronin, "Operads and varieties of algebras defined by polylinear identities," Sib. Math. J. **47** (3), 555–573 (2006).
14. S. N. Tronin, "Natural multitransformations of multifunctors," Russian Math. **55** (11), 49–60 (2011).
15. A. Gaynullina, "On one class of commutative operads," Asian-European J. Math. **10** (1), 1 750 007 (2017).
16. A. G. Pinus, *Conditional Terms and Their Applications in Algebra and Computation Theory* (Novosibirskii Gos. Tekhn. Univ., Novosibirsk, 2002) [in Russian].
17. D. Grigoriev and V. Shpilrain, "Tropical cryptography," Communication in Algebra **42**(6), 2624–2632 (2014).